



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-452111

Simulation of Intense Beams and Targets for Heavy-Ion-Fusion Science (HEDLP / Inertial Fusion Energy)

A. Friedman, J. J. Barnard, R. H. Cohen, M. Dorf, D. Eder, D.
P. Grote, S. M. Lund, W. M. Sharp, E. Henestroza, E. P. Lee,
J. L. Vay, R. C. Davidson, I. D. Kaganovich, H. Qin, E.
Startsev, K. Fagnan, A. Koniges, A. Bertozzi

August 31, 2010

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

1.1.1. Simulation of intense beams and targets for heavy-ion-fusion science (HEDLP / Inertial Fusion Energy)

PI: Alex Friedman, LLNL, LBNL, and Heavy Ion Fusion Science Virtual National Laboratory

Contributors: John Barnard, Ron Cohen, Mikhail Dorf, David Eder, Dave Grote, Steve Lund, Bill Sharp, LLNL; Enrique Henestroza, Ed Lee, Jean-Luc Vay, LBNL; Ron Davidson, Igor Kaganovich, Hong Qin, Ed Startsev, PPPL; Kirsten Fagnan, Alice Koniges, NERSC; Andrea Bertozzi, UCLA

Repository: mp42

1.1.1.1. Summary and Scientific Objectives

The US Heavy Ion Fusion Science Virtual National Laboratory (HIFS-VNL), a collaboration of LBNL, LLNL, and PPPL, conducts research on the science of intense heavy-ion beams, on high-energy-density physics (HEDP) and especially Warm Dense Matter (WDM) generated by ion beams, and on target physics for ion-beam-driven Inertial Fusion Energy. Ongoing experiments are therefore focused on generating, compressing, and focusing space-charge-dominated beams and using them to heat thin foils, the evolution and properties of which are then measured. To further this work, a new accelerator facility, the Neutralized Drift Compression Experiment-II (NDCX-II), is under construction at LBNL, with completion planned for 2012. Obtaining maximum benefit from these experiments is a key near-term goal of the simulation program. Recent simulation efforts in support of NDCX-II have concentrated on developing the physics and engineering design of the NDCX-II accelerator, on identifying favorable operating points, and on planning the Warm Dense Matter experiments to be done with its beam once operations ensue. As we transition to support of the actual experiments, the scope of our beam simulations must expand greatly, with primary emphasis on detailed simulations of the actual beam as realized and its interaction with the target. This will include extensive studies of the coupled dynamics of the beam and the neutralizing plasma (which allows the beam to be compressed to a compact volume in space and time), and routine transfer of the ion beam data at the target plane from our main beam physics code, Warp, into hydrodynamic simulations of the target behavior using the Hydra code (run at LLNL) and the new ALE-AMR code (run at NERSC).

Intense ion beams are non-neutral plasmas and exhibit collective, nonlinear dynamics that must be understood using the kinetic models of plasma physics. This physics is rich and subtle: a wide range in spatial and temporal scales is involved, and effects associated with instabilities and non-ideal processes must be understood. In addition, multispecies effects must be modeled during the interaction of the beams with any stray electrons in the accelerator and with the neutralizing plasma in the target chamber. The models must account for the complex set of interactions among the various species, with the walls, and with the applied and self-fields. Finally, oscillations of the beam core and “mismatches” of the beam confinement can dilute the beam phase space and parametrically pump particles into a low-density outlying “halo” population; this physics imposes stringent requirements on numerical noise, requiring good particle statistics and mesh resolution. A blend of numerical techniques, centered around the Particle-In-Cell method, are used in Warp to address these needs, including: electrostatic and electromagnetic solvers, adaptive mesh refinement, cut-cell boundaries, large time step particle pusher, and implicit field solvers and particle pushers. Ion beams have a long memory, and initialization of a simulation at mid-system with an idealized particle distribution is often unsatisfactory; thus, a key goal is to further develop and extensively exploit an integrated and detailed source-to-target beam simulation capability.

In order to determine material properties (such as equation of state) in the warm dense matter (WDM) regime, we must use simulation codes to help interpret experimental diagnostics. The WDM regime is at the boundaries of solid state physics and plasma physics, between non-degenerate material and degenerate, between ionized and neutral, between liquid and vapor. To understand and simulate the results of experiments that volumetrically heat material, many processes must be combined into a single simulation. These include phase changes, ionization processes, shock processes, spall processes,

and droplet formation, to name a few. Our goal is to fully characterize the experiments that will be conducted on NDCX-II. This accelerator-based facility will compress ion beams and use them to heat material targets to temperatures of about 10,000 degrees K. The material is heated so rapidly that, although its temperature is well above the vaporization temperature, its inertia will keep it at solid density (at least for an inertial confinement time of order one ns). Experimental diagnostics will record the response of the material (e.g. temperature, density, velocity) to infer the equation of state and other properties. Synthetic diagnostics in the simulations will be essential for inter-comparison.

Hydrodynamic codes such as Hydra and ALE-AMR can be used to model ion deposition and the subsequent response of the target to bulk heating. The ALE-AMR code can also model the strength and failure of materials using various models, some of which include history variables and non-isotropic terms. We are implementing surface tension effects in ALE-AMR. The formation of droplets has been one of the dominant features of data from the predecessor to NDCX-II, NDCX-I, yet there are no available codes that can describe the interplay between surface tension, inertial, and van der Waals forces that form the droplets. The droplets are predicted to be on the 0.1 micron scale for NDCX-II, yet the gradient scale length for the ion deposition (i.e. the transverse focal spot size) is on the 500 micron scale. So our intent is to model the formation of droplets initially on a microscopic patch of the target, but eventually increase the computational domain of the simulation until the entire beam heated part of the foil is included in the simulation.

Within 3 to 5 years, we expect that end-to-end simulations of the beam from source through target, and of the detailed target response, will be routinely carried out at high fidelity. We will also be exploring extensions to NDCX-II, and/or a follow-on facility, requiring extensive use of ensembles of runs, as has been done for the baseline design of NDCX-II.

Our principal goals, and activities in support of those goals, over the next five years are as follows:

(1) Optimize the properties of the NDCX-II beam for each class of target experiments; achieve quantitative agreement with measurements; develop improved machine configurations and operating points. To accomplish these goals, we plan to use Warp to simulate NDCX-II from source to target, in full kinetic detail, including first-principles modeling of beam neutralization by plasma. The output from an ensemble of Warp runs (representing shot-to-shot variations) will be used as input to target simulations using ALE-AMR on NERSC, and other codes.

(2) Develop enhanced versions of NDCX-II (the machine is designed to be extensible and reconfigurable), and carry out studies to define a next-step ion beam facility. To accomplish these goals, much of the work will involve iterative optimization employing Warp runs that assume ideal beam neutralization downstream of the accelerator.

(3) Carry out detailed target simulations in the Warm Dense Matter regime using the ALE-AMR code, including surface tension effects, liquid-vapor coexistence, and accurate models of both the driving beam and the target geometry. For this we will need to make multiple runs (to capture shot-to-shot variations), and to both develop and employ synthetic diagnostics (to enable comparison with experiments). The new science that will be revealed is the physics of the transition from the liquid to vapor state of a volumetrically superheated material, wherein droplets are formed, and wherein phase transitions, surface tension and hydrodynamics all play significant roles in the dynamics. These simulations will enable calculations of equation of state and other material properties, and will also be of interest for their illumination of the science of droplet formation.

1.1.1.2. Methods of Solution

Our main ion-beam code, Warp, was originally to simulate space-charge-dominated beam dynamics in induction accelerators for heavy-ion fusion (HIF). In recent years, the physics models in the code have been generalized, so that Warp can model beam injection, complicated boundary conditions, denser plasmas, a wide variety of accelerator “lattice” components, and the non-ideal physics of beams interacting with walls and plasmas. The code now has an international user base and is being applied to projects both within and far removed from the HIF community.

Warp uses a flexible multi-species particle-in-cell model to describe beam dynamics and the electrostatic or electromagnetic fields in particle accelerators, particularly those driven by induction modules. While the core routines of Warp solve finite-difference representations of Maxwell's equations and relativistic or non-relativistic motion equations, the code also uses a large collection of subordinate models to describe lattice elements and such physical processes as beam injection, desorption, and ionization. The representation of particles by a much smaller number of "macroparticles" can be derived from Boltzmann's equation, describing the evolution of a population of particles interacting by collisions and the collective fields.

Warp is a 3-D time-dependent multiple-species particle-in-cell (PIC) code, with the addition of a Warped-coordinate particle advance to treat particles in a curved beam pipe. Self-fields are obtained via Poisson equations for the scalar and vector potentials or full electromagnetic via Maxwell equations. Simplified models are available for the self-magnetic and inductive forces. Time-dependent applied external fields can be specified through the Python user interface. Warp also has 2-D models, using Cartesian or cylindrical geometry, as well as a module representing the beam with a 4-D Vlasov formulation and with low-order moment equations. Models are available for background gas, wall effects, stray electrons, space-charge-limited and source-limited emission, and atomic processes such as charge exchange. Elaborate initialization and run-time options allow realistic modeling of induction accelerators. A beam may be initialized with one of many analytic distributions or with a distribution synthesized from experimental data, or ions can be emitted from a flat or curved diode surface. Lattice-element fields may be represented by several options, from simple hard-edge analytic forms to first-principles 3-D calculations. Poisson's equation can be solved using several methods, including FFT, Multigrid, and AMR/Multigrid. The electromagnetic (EM) solver can also use MR. With multigrid, the Shortley-Weller method for the subgrid-resolution description of conductors allows the use of complicated boundary conditions.

Parallelization of Warp is done using domain decomposition with MPI. Warp uses independent spatial decompositions for particles and field quantities, allowing the particle and field advances to be load-balanced independently; we recently added more general decompositions. In transverse-slice 2-D runs, the field solution is repeated on each node, but solved in parallel by processors within a node.

The size and duration of Warp jobs varies tremendously, depending on such factors as problem dimensionality, grid size, duration, particle count, and the physical processes being modeled. However, with our generalized decomposition, we do not foresee any limitation resulting from the code's architecture. For a 3-D test problem using 512x512x512 cells, we have demonstrated excellent parallel scaling of the electromagnetic PIC capability, up to nearly 35,000 processors.

Our Warp projects tend not to be data intensive, they use modest amounts of memory but require many time steps. We typically run (in 2-D or 3-D) with of order 100 grid cells along each axis. Higher resolution and large 3-D simulations typically have a mesh of order 100s by 100s by 1000s of grid cells. The data per cell is either a single point or a 3-D vector. Typically of order 1,000,000 particles are used, with 13 or more variables per particle. We currently use 512 to 1024 processors for typical Franklin runs and 4096 for a few key runs with fine grids and an augmented number of particles.

ALE-AMR is a relatively new code that combines Arbitrary Lagrangian Eulerian (ALE) hydrodynamics with Adaptive Mesh Refinement (AMR) to connect the continuum to micro-structural regimes. The code is unique in its ability to model both hot radiating plasmas and cold fragmenting solids. The hydrodynamics are done in a Lagrangian model (wherein material moves with the mesh), but the resulting mesh can be modified to prevent tangling or severe mesh distortions. If the entire mesh is restored to the mesh of the previous time-step after every step, the code is said to be run in Eulerian mode (fixed mesh). In general, this is not done, and we only modify a portion of the mesh during a fraction of the time steps. This ability to do selective remapping is the reason to use the word "arbitrary." We also employ the Hydra code, another 3-D radiation hydrodynamics ALE code; that code is run on LLNL computers, which are accessed from LBNL and LLNL by group members. A common feature of ALE codes is the ability to have multiple materials in a given computational zone. Such mixed zones are generally created during the advection phase of the advance, when material

from the old mesh is transferred to the new mesh. The ALE-AMR code uses a volume-of-fluids approach to calculate the interface between different materials in a zone. Information from neighboring zones can be used to explicitly construct the interfaces if needed.

One key added capability of ALE-AMR, relative to other ALE codes such as Hydra, is the ability to dynamically add mesh elements (refinement) or remove mesh elements (coarsening) during the run. This capability is called Adaptive Mesh Refinement (AMR); however, the ability to remove zones by coarsening the mesh when there are no longer any steep gradients is also important. ALE-AMR refines by a factor of three along each dimension, so in 3D one zone becomes 27 zones. During refinement all material interfaces must be explicitly defined to place the correct amount of each material in the new zones. Numerical techniques were developed for many of the physics packages to work efficiently on a dynamically moving and adapting mesh. ALE-AMR also continues several features that allow for very long-time simulations, a unique fragmentation capability, and the ability to “shape-in” unusual objects.

Additional physics, beyond basic hydrodynamics, is implemented in ALE-AMR using operator splitting. For example, a flexible strength/failure framework allows “pluggable” material models to update the anisotropic stress tensor that is used in the hydro advance during the following step. The code also includes an ion deposition model for bulk heating of the material, and both heat conduction and radiation transport using the diffusion approximation. The hydro uses explicit time stepping but some packages, e.g., radiation transport, can do an implicit solve at each explicit time step.

The parallelism in ALE-AMR is currently MPI-only with the ability to do dynamic load balancing based on the computational requirements. The domain decomposition is zonal. During the ion deposition phase of the simulation, the regions with ion beams will have smaller number of zones in the domain assigned to a given processor because of the additional computation work associated with beam energy deposition. There are various places in the code where additional levels of parallelism are possible and we are investigating hybrid models, e.g., OpenMP + MPI.

1.1.1.3. HPC Requirements

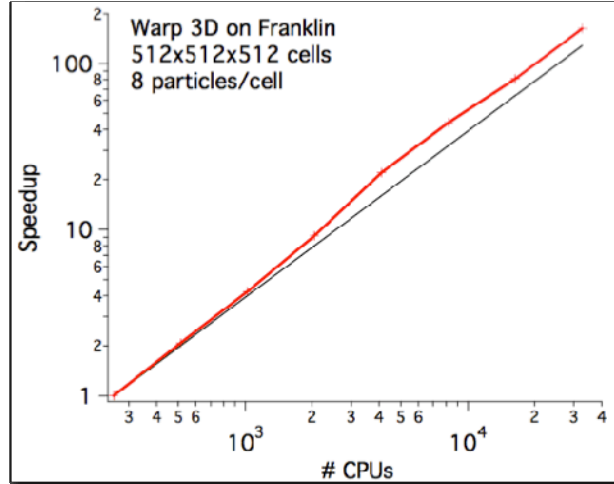
Our need for NERSC time is rapidly growing. In past years our requirements were modest. We concentrated on NDCX-II machine design using simplified tools such as a 1-D beam physics code (ASP), and on Warm Dense Matter simulations using both Hydra (at LLNL) and a specialized 1-D code (Dish). Recently, we began applying NERSC resources to iterative design calculations for the NDCX-II facility, and our usage rate increased roughly five-fold. A NISE allocation of 500,000 hours has been extremely valuable. Now, two developments compel us to carry out far more demanding simulations at NERSC: (1) the need to capture beam-in-plasma effects (requiring far more simulation particles, and a smaller time-step size and grid spacing); and (2) the introduction of the ALE-AMR code into our group (requiring considerable resources for realistic problems).

We begin with a discussion of recent Warp usage, which has emphasized iterative design and assessment (on NERSC and LBNL clusters) using ensembles of runs with random errors. For this task 256 cases (instances) are typically run in a single batch job, 8 at a time. This employs 128 cores, with less than 1 GB/core, using 60 GB total memory and 35 hours of wall-clock time. Much data processing is in-line, and I/O is only about 100 GB / batch job. This approach leads to very light traffic in and out of NERSC, with results stored at the Center.

Another class of Warp runs models ion beams in plasmas. *Current* problems of this type use 100's x 100's x 1000's of cells, and millions of particles (with 13 or more variables per particle), with 512, 1024, and sometimes 4096 processors on Franklin. With the electromagnetic Maxwell (EM) field model, tests show good scaling at fixed problem size (512³ cells) to 35,000 processors (see Figure).

We project the need for four classes of Warp runs during the next five years: (1) Ensemble runs to optimize the output beam from the NDCX-II *accelerator*, for each class of target being shot. So far, we haven't used gradient methods for optimization because of particle noise; we hope to overcome this with larger runs. (2) Simulations of plasma injection into the drift-compression line and final-

focus solenoid, which can be quite costly because the plasma flow is relatively slow (~ 10 ms) and it is necessary to operate on an electron timescale. Both EM and explicit electrostatic (ES) models are used; run times are comparable because the time-step size in the ES model, which is set by the need to resolve plasma oscillations, is near the Courant limit for light waves on the mesh. Also, the EM algorithm scales more readily to very large numbers of processors. (3) Integrated simulations of one or more beams compressing in a neutralizing plasma (with properties obtained from plasma-injection runs as described above, or via measurements). Such runs require less computer time than plasma injection runs, because the beam is in the system for < 1 ms; however, ensembles are typically needed. (4) Detailed simulations resolving short time- and space-scales for, *e.g.*, two-stream instability. Since the highest growth rates for a cold beam and plasma are for short wavelengths, while we seek to capture the overall system scale, such runs can be costly, even in axisymmetric (r, z) geometry.



1.1.1.4. Computational and Storage Requirements Summary

| Total (both main codes) | Current usage | Projected 2014 |
|-------------------------|---------------|----------------|
| Computational Hours | 0.6 M | 60 M |

The “current usage” numbers above include about 200k hours / year on the Fusion and Lawrence Livermore clusters, and an estimated 400k hours at NERSC (twice our usage during the first half of 2010). Our NERSC usage in 2009 was 74k hours. A NISE allocation of 500k hours has been extremely useful.

| Warp code (ion beam simulation) | Current usage | Projected 2014 |
|---------------------------------|---------------|------------------|
| Computational Hours | 0.5 M | 40 M |
| Parallel Concurrency | 128 – 32,000 | 50,000 – 500,000 |
| Wall Hours per Run | 2 – 35 | 10 – 100 |
| Aggregate Memory | 60 GB | 3 – 30 TB |
| Memory per Core | 1 GB | 1 GB |
| I/O per Run | 0.1 TB | 0.5 – 5 TB |
| On-Line Storage | 1 TB | 10 – 100 TB |
| Data transfer in/out of NERSC | 0.1 TB / year | 1 TB / year |

A proper estimate of the computer resources required for an integrated, end-to-end kinetic simulation of beam(s) and plasma using Warp would assess the several regions of the system separately and would assume use of AMR, variable time-step sizes, and perhaps high-order differencing (to allow the cell size Δx to exceed the Debye shielding length λ_D by large factors). Such an estimate would itself require the development and use of a small computer program.

| ALE-AMR code (target simulation) | Current usage | Projected 2014 |
|---|---------------|-----------------|
| Computational Hours | 0.1 M | 20 M |
| Parallel Concurrency | 128 – 10,000 | 10,000 – 30,000 |
| Wall Hours per Run | 6 – 50 | 20 – 150 |
| Aggregate Memory | 64 – 5,000 GB | 5 – 15 TB |
| Memory per Core | 1 GB | 1 GB |
| I/O per Run | 0.2 TB | 1 – 10 TB |
| On-Line Storage | 2 TB | 20 – 200 TB |
| Data transfer in/out of NERSC | 0.1 TB / year | 1 TB / year |

The computational requirements for target simulations using ALE-AMR are large because of the wide range of spatial scales involved and the need to include multiple detailed physics models. The width of the ion beam is generally of order 0.5 mm, which requires a computational mesh of a few mm in each dimension. However, droplets as small as 0.1 micron are expected, and it is desired to have a number of zones within each droplet. For a uniform 3D run, this would imply the need for an unacceptable number ($\sim 10^{12}$ - 10^{15}) of zones. The ratio of the zone size at the coarsest level to the zone size at the finest level is greater than 10^7 for 6 levels of refinement. Even with this level of refinement, the number of zones in the coarse level would still be too large to model the entire domain. Only selected regions will be modeled at the highest resolution with the resulting information (e.g., the droplet distribution) being assumed to be relevant to other regions with similar hydrodynamic properties (i.e., pressure, density, temperature, stress, and strain rates).

1.1.1.5. Support Services and Software

For some of our applications, long batch queues and limitations on interactive use are impeding factors. Interactivity is useful for debugging and for diagnostics development, because Warp produces many of its diagnostics on-line (thereby avoiding massive data transfers, offloading mostly processed data). The Lawrence Livermore and Fusion clusters at LBNL have been convenient, especially for multi-week runs, but are limited in capacity, capability, and availability.

The Warp code is deeply tied to Python. Dynamic libraries are part of the core of Python and it would be useful for them to be supported. This would greatly ease code maintenance and preserve flexibility. Dynamic libraries are not actually essential, since Warp and Python can be built statically. However, static loading requires modification of the Python source, introduces a more complicated and fragile build process, and makes installing new and upgrading existing packages more difficult since they must be manually incorporated into the build system (when they could otherwise be installed independently). A secondary need, related to the startup time, is an efficient mechanism for Python on the nodes to read in the sizable number of startup scripts and dynamic objects (when supported).

Scaling studies of ALE-AMR, showed excellent parallel scaling to >5000 cores, but uncovered memory access differences on Franklin during initialization. For example, when the large equation-of-state table was moved from /project to /home there was a 10X speedup, with further speedup observed when the table was read from \$SCRATCH (this is the current mode of operation). These effects are

still being studied, but they do not impact the performance of the code after initialization.

1.1.1.6. Emerging HPC Architectures and Programming Models

Warp's core loop incorporates the PIC elements: field solve, gathering of field onto particles, particle advance, charge or current deposition onto grid. The algorithms that have been implemented to enable progressing toward high fidelity end-to-end calculations span the space between (a) "computationally intensive" massively-parallel methods using the lowest level of approximation with explicit solvers on uniform grids, small time steps, and large numbers of macroparticles, and (b) "algorithmically intensive" moderately-parallel methods using higher-level approximations involving implicit methods, global solvers, AMR grids, large time steps, and moderate numbers of macroparticles.

We will pursue both efficiency and flexibility so that code users may continue to employ varying levels of description, itself a key to insight. On a five-year timescale we must accommodate a significant increase in cores per node, a decrease in memory per core, and an increase in the computational cost of data movement. At this point it is unclear whether the NERSC-7 computer will employ a descendant of today's CPU/GPU architecture and programming model, a Blue-Gene-like configuration, or an intermediate model. It is very possible that one combination of algorithms will perform better on one machine while a different combination will be more adapted to another.

In any case, data locality will be important. The gathering of fields, particle advance, and charge or current deposition are local if properly implemented. A complication is the need for scatter-add deposition of source terms from the particles onto the mesh. Integrity of these terms must be preserved as we optimize performance using (probably) MPI plus on-node parallelism, and AMR.

The explicit electromagnetic (Maxwell) field advance (Finite-Difference Time-Domain) is also local. However, the electrostatic field model is based on a global elliptic solver (multigrid is our main approach) is likely to be more challenging; for it we plan to pursue optimization of our own AMR/multigrid solver, since it interfaces nicely with Warp's data structures, but also will assess other available packages as they are developed. Much of our work to date has used the electrostatic field model, which is appropriate for many of our problems. However, we may shift emphasis to an explicit electromagnetic Maxwell-Vlasov formulation. Given the denser neutralizing plasmas anticipated in future experiments, the time-step size for EM is likely to be similar to that for ES.

We will partner with others in the particle-in-cell simulation community and the computer science community, to develop, test, and implement multi-level decompositions that make effective use of the new architectures that emerge. We plan to extract kernels of the main components and analyze their projected performance on those architectures, and select the most effective methods.

Initial efforts at porting PIC methods to GPU architectures have been encouraging. However, while our beams are non-neutral or neutralized plasmas, they are not homogeneous plasmas, and boundary condition handling can be nontrivial. We must also deal with collision events (between charged particles, and with neutrals), and must collect more extensive particle diagnostics than on most other PIC codes. These attributes will need to be accommodated.

For the ALE-AMR code, we will also carry out the necessary optimizations to enable efficient use of the NERSC-7 machine. At present the code employs the SAMRAI AMR package, but it could as readily employ the CHOMBO package should that framework be more efficient on NERSC-7. In any event, performance will depend heavily on that of the underlying package. Physics such as surface tension, important to understanding the formation of droplets in NDCX, may be implemented through operator splitting techniques, allowing some degree of independent optimization.